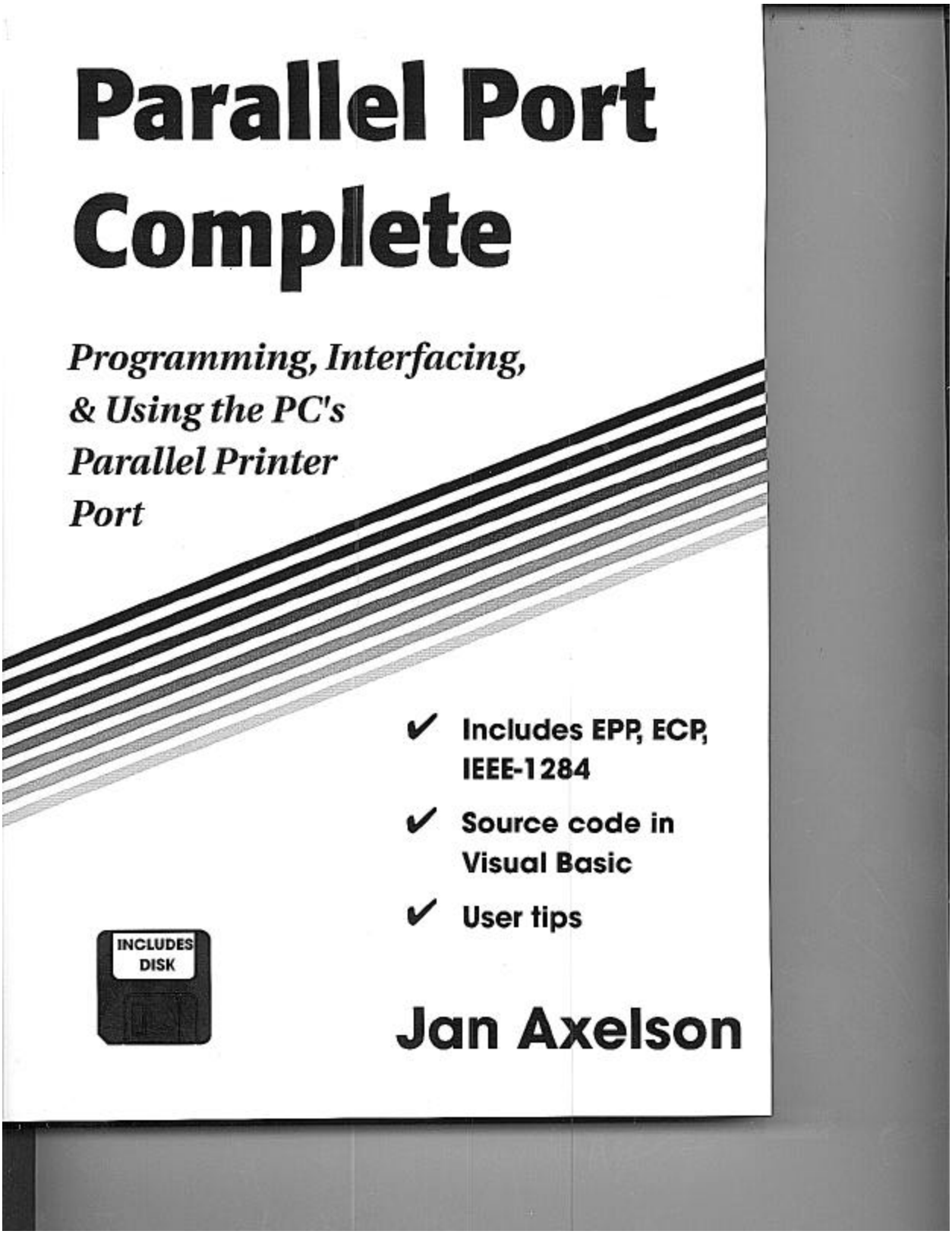
Jameco Part number 138261


Parallel Port Complete
Programming, Interfacing, & Using the PC's Parallel Printer Port
✔ Includes EPP, ECP, IEEE-1284
✔ Source code in Visual Basic
✔ User tips
INCLUDES DISK
Jan Axelson

Jameco Part number 138261

*PC hardware & programming*

# Parallel Port Complete

## Get the most from the Versatile interface that's Built into every PC

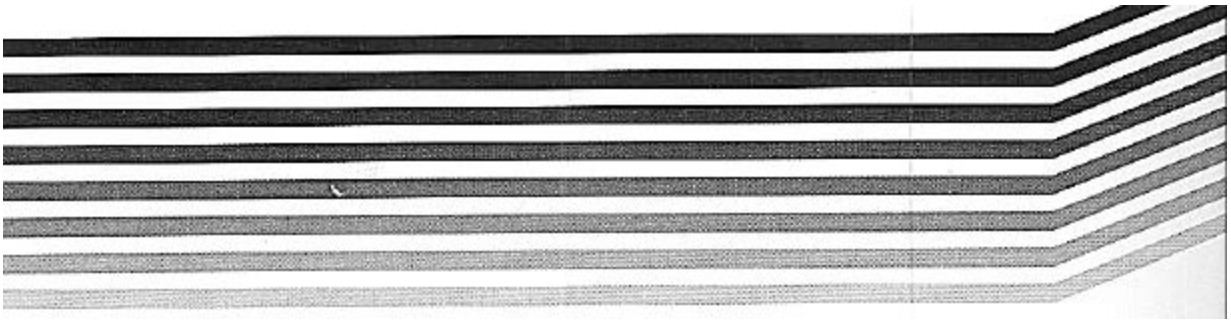
**This book is for programmers, hardware designers, and anyone who uses the PC's parallel port to communicate with printers and other peripheral devices. The tips, tools, and examples in this complete reference will save you time, spark new ideas for your own projects, and help you use all of a port's abilities--including the new high-speed, bidirectional modes.**

**Author Jan Axelson shows how to:**

- **Find, configure, and test ports, including the new enhanced (EPP), extended-capabilities (ECP), and simple bidirectional (PS/2-type) ports**
- **Transfer data using the IEEE-1284 standard's Compatibility, Byte, EPP, and ECP modes**
- **Access ports in Visual Basic under Windows 3.1 and Windows 95**
- **Select cables for faster transfers and longer links**
- **Design programs and hardware interfaces that will work with any port**
- **Use parallel ports to connect two PCs**

**Use these included tools and examples for a quick start to your projects:**

- **DLLs for port access in Visual Basic**
- **Visual Basic forms and routines to perform common parallel-port functions**
- **Circuit diagrams and Visual-Basic source code for these applications: simple and latched inputs and outputs - input and output expansion - switching power to a load - controlling a switch matrix - measuring analog signals - causing periodic events - triggering on external events - Compatibility, Nibble, Byte, EPP, and ECP mode peripheral interfaces - and much more!**



**Jan Axelson is the author of *The Microcontroller Idea Book* and *Making Printed Circuit Boards*. Jan's simple, direct writing style and focus on practical details have made her column in *The Microcomputer Journal* a favorite of readers for many years.**

ISBN 0965081917

9 780965 081917

# Table of Contents


| | | |
|---|---|---|
| | **Introduction** | **ix** |
| **1** | **Essentials** | **1** |
| | **Defining the Port** | **1** |
| | Port Types | |
| | **System Resources** | **4** |
| | Addressing | |
| | Interrupts | |
| | DMA Channels | |
| | Finding Existing Ports | |
| | **Configuring** | **6** |
| | Port Options | |
| | Drivers | |
| | Adding a Port | |
| | **Port Hardware** | **9** |
| | Connectors | |
| | The Circuits Inside | |
| | Cables | |
| | **Multiple Uses for One Port** | **11** |
| | Security Keys | |
| | **Alternatives to the Parallel Port** | **13** |
| | Serial Interfaces | |

Other Parallel Interfaces
Custom I/O Cards
PC Cards

**2 Accessing Ports 17**

**The Signals 17**
Centronics Roots
Naming Conventions
The Data Register
The Status Register
The Control Register
Bidirectional Ports

**Addressing 24**
Finding Ports

**Direct Port I/O 26**
Programming in Basic
Other Programming Languages

**Other Ways to Access Ports 31**
LPT Access in Visual Basic
Windows API Calls
DOS and BIOS Interrupts

**3 Programming Issues 39**

**Options for Device Drivers 39**
Simple Application Routines
DOS Drivers
Windows Drivers
Custom Controls

**Speed 45**
Hardware Limits
Software Limits

**4 Programming Tools 53**

**Routines for Port Access 53**
Data Port Access
Status Port Access
Control Port Access
Bit Operations

**A Form Template 60**
Saving Initialization Data
Finding, Selecting, and Testing Ports

**5 Experiments 85**

**Viewing and Controlling the Bits 85**
Circuits for Testing
Output Types
Component Substitutions

**Cables & Connectors for Experimenting** **99**
**Making an Older Port Bidirectional** **100**
Cautions
The Circuits
The Changes
**6 Interfacing** **105**
**Port Variations** **105**
Drivers and Receivers
Level 1 Devices
Level 2 devices
**Interfacing Guidelines** **110**
General Design
Port Design
**Cable Choices** **112**
Connectors
Cable Types
Ground Returns
36-wire Cables
Reducing Interference
Line Terminations
Transmitting over Long Distances
**Port-powered Circuits** **124**
When to Use Port Power
Abilities and Limits
Examples
**7 Output Applications** **129**
**Output Expansion** **129**
**Switching Power to a Load** **132**
Choosing a Switch
Logic Outputs
Bipolar Transistors
MOSFETs
High-side Switches
Solid-state Relays
Electromagnetic Relays
Controlling the Bits
X-10 Switches
**Signal Switches** **143**
Simple CMOS Switch
Controlling a Switch Matrix
**Displays** **148**
**8 Input Applications** **149**
**Reading a Byte** **149**

Latching the Status Inputs
Latched Input Using Status and Control Bits
5 Bytes of Input
Using the Data Port for Input
**Reading Analog Signals** **154**
Sensor Basics
Simple On/Off Measurements
Level Detecting
Reading an Analog-to-digital Converter
Sensor Interfaces
Signal Conditioning
Minimizing Noise
Using a Sample and Hold
**9 Synchronous Serial Links** **165**
**About Serial Interfaces** **165**
**A Digital Thermometer** **166**
Using the DS1620
The Interface
An Application
Other Serial Chips
**10 Real-time Control** **183**
**Periodic Triggers** **183**
Simple Timer Control
Time-of-day Triggers
Loop Timers
**Triggering on External Signals** **189**
Polling
Hardware Interrupts
Multiple Interrupt Sources
Port Variations
**11 Modes for Data Transfer** **203**
**The IEEE 1284 Standard** **203**
Definitions
Communication modes
**Detecting Port Types** **207**
Using the New Modes
Port Detecting in Software
Disabling the Advanced Modes
**Negotiating a Mode** **210**
Protocol
**Controller Chips** **212**
Host Chips
Peripheral Chips
Peripheral Daisy Chains

**Programming Options** 220
**12 Compatibility and Nibble Modes** 223
**Compatibility Mode** 223
Handshaking
Variations
**Nibble Mode** 228
Handshaking
Making a Byte from Two Nibbles
**A Compatibility & Nibble-mode Application** 232
About the 82C55 PPI
Compatibility and Nibble-mode Interface
**13 Byte Mode** 249
**Handshaking** 249
**Applications** 250
Compatibility & Byte Mode
Compatibility, Nibble & Byte Mode with Negotiating
**14 Enhanced Parallel Port: EPP** 267
**Inside the EPP** 267
Two Strobes
The Registers
**Handshaking** 269
Four Types of Transfers
Switching Directions
Timing Considerations
**EPP Variations** 275
Use of nWait
Clearing Timeouts
Direction Control
**An EPP Application** 277
The Circuit
Programming
**15 Extended Capabilities Port: ECP** 285
**ECP Basics** 286
The FIFO
Registers
Extended Control Register (ECR)
Internal Modes
**ECP Transfers** 289
Forward transfers
Reverse Transfers
Timing Considerations
Interrupt Use

Using the FIFO
**Other ECP Modes** **296**
Fast Centronics
Test Mode
Configuration Mode
**An ECP Application** **298**
**16 PC-to-PC Communications 305**
**A PC-to-PC Cable** **305**
**Dos and Windows Tools** **306**
MS-DOS's Interlnk
Direct Cable Connection
**A PC-to-PC Application** **311**
**Appendices**
**A Resources** **323**
**B Microcontroller Circuit** **327**
**C Number Systems** **329**
**Index** **333**

# Introduction

From its origin as a simple printer interface, the personal computer's parallel port has evolved into a place to plug in just about anything you might want to hook to a computer. The parallel port is popular because it's versatile--you can use it for output, input, or bidirectional links--and because it's available--every PC has one.

Printers are still the most common devices connected to the port, but other popular options include external tape and disk drives and scanners. Laptop computers may use a parallel-port-based network interface or joystick. For special applications, there are dozens of parallel-port devices for use in data collection, testing, and control systems. And the parallel port is the interface of choice for many one-of-a-kind and small-scale projects that require communications between a computer and an external device.

In spite of its popularity, the parallel port has always been a bit of a challenge to work with. Over the years, several variations on the original port's design have emerged, yet there has been no single source of documentation that describes the port in its many variations.

I wrote this book to serve as a practical, hands-on guide to all aspects of the parallel port. It covers both hardware and software, including how to design external

circuits that connect to the port, as well as how to write programs to control and monitor the port, including both the original and improved port designs.

**Who should read this book?**

The book is designed to serve readers with a variety of backgrounds and interests:

**Programmers** will find code examples that show how to use the port in all of its modes. If you program in Visual Basic, you can use the routines directly in your programs.

For **hardware designers,** there are details about the port circuits and how to interface them to the world outside the PC. I cover the port's original design and the many variations and improvements that have evolved. Examples show how to design circuits for reliable data transfers.

System **troubleshooters** can use the programming techniques and examples for finding and testing ports on a system.

**Experimenters** will find dozens of circuit and code examples, along with explanations and tips for modifying the examples for a particular application.

**Teachers and students** have found the parallel port to be a handy tool for experiments with electronics and computer control. Many of the examples in this book are suitable as school projects.

And last but not least, **users,** or anyone who uses a computer with printers or other devices that connect to the parallel port, will find useful information, including advice on configuring ports, how to add a port, and information on cables, port extenders, and switch boxes.

**What's Inside**

This book focuses on several areas related to the parallel port:

**Using the New Modes**

Some of the most frequently asked parallel-port questions relate to using, programming, and interfacing the port in the new, advanced modes, including the enhanced parallel port (EPP), the extended capabilities port (ECP), and the PS/2-type, or simple bidirectional, port. This book covers each of these. Examples show how to enable a mode, how to use the mode to transfer data, and how to use software negotiation to enable a PC and peripheral to select the best mode available.

### Visual Basic Tools

Microsoft's Visual Basic is one of the most popular programming languages for PCs, and this book includes programming tools to help in writing Visual-Basic programs that access the parallel port. One tool is a Visual-Basic form that enables users to find, select, and test the parallel ports on a system. You can use the form as a template, or beginning form, for applications you write. Also included is a set of routines that simplify reading and writing to the parallel port's registers and reading and changing individual bits in a byte.

Because Visual Basic doesn't include functions for performing simple reads and writes to a port, the companion disk included with this book includes DLLs that add these abilities to Visual Basic. Versions are included for use with both 16-bit and 32-bit programs.

### Applications

Besides the general-purpose programming tools, I've included a variety of example circuits with Visual-Basic code for controlling and monitoring external circuits. The examples include popular applications such as switching power to a load, reading analog signals, expanding the number of inputs and outputs that the port can access, and interfacing to a microcontroller circuit. One example shows how to use the parallel port to communicate with chips that use a synchronous serial interface. A chapter on real-time control shows how to write programs that trigger on external events, such as a signal transition at the parallel port or time or calendar information. There's a discussion and examples of using the parallel port as the power source for low-power external circuits.

### Cables and Interfacing

The proper cable can mean the difference between a link that works reliably and one that doesn't. This book shows how to choose an appropriate parallel-port cable, and how to design the circuits that interface to the cable.

### PC-to-PC Communications

Although the parallel port was originally intended as an interface between a PC and a printer or other peripheral, it's also become a popular interface for transferring information between two PCs. This book shows how to set up a PC-to-PC link using the parallel ports and either the operating system's built-in tools or your own programs.

Jameco Part number 138261

Introduction

**About the Program Code**

Every programmer has a favorite language. The choices include various implementations of Basic, C/C++, and Pascal/Delphi, and assembly language.

For the program examples in this book, I wanted to use a popular language so as many readers as possible could use the examples directly, and this prompted my decision to use Microsoft's Visual Basic for Windows. A big reason for Visual Basic's popularity is that the programming environment makes it extremely easy to add controls and displays that enable users to control a program and view the results.

However, this book isn't a tutorial on Visual Basic. It assumes you have a basic understanding of the language and how to create and debug a Visual-Basic program.

I developed the examples originally using Visual Basic Version 3, then ported them to Version 4. As much as possible, the programs are designed to be compatible with both versions, including both 16- and 32-bit Version-4 programs. The companion disk includes two versions of each program, one for Version 3 and one for 16- and 32-bit Version 4 programs.

One reason I decided to maintain compatibility with Version 3 is that the standard edition of Version 4 creates 32-bit programs only. Because Windows 3.1 can't run these programs, many users haven't upgraded to Version 4. Also, many parallel-port programs run on older systems that are put to use as dedicated controllers or data loggers. Running the latest version of Windows isn't practical or necessary on these computers.

Of course, in the software world, nothing stays the same for long. Hopefully, the program code will remain compatible in most respects with later versions of Visual Basic.

Compatibility with Version 3 does involve some tradeoffs. For example, Version 3 doesn't support the Byte variable type, so my examples use Integer variables even where Byte variables would be appropriate (as in reading and writing to a byte-wide port). In a few areas, such as some Windows API calls, I've provided two versions, one for use with 16-bit programs, Version 3 or 4, and the other for use with Version 4 programs, 16- or 32-bit.

In the program listings printed in this book, I use Visual Basic 4's line-continuation character (_) to extend program lines that don't fit on one line on the page. In other words, this:

```
PortType =    _
Left$(ReturnBuffer, NumberOfCharacters)
```

is the same as this:

```
PortType = Left$(ReturnBuffer, NumberOfCharacters)
```

To remain compatible with Version 3, the code on the disk doesn't use this feature.

Most of the program examples are based on a general-purpose Visual-Basic form and routines introduced early in the book. The listings for the examples in each chapter include only the application-specific code added to the listings presented earlier. The routines within a listing are arranged alphabetically, in the same order that Visual Basic displays and prints them.

Of course, the concepts behind the programs can be programmed with any language and for any operating system. In spite of Windows' popularity, MS-DOS programs still have uses, especially for the type of control and monitoring programs that often use the parallel port. Throughout, I've tried to document the code completely enough so that you can translate it easily into whatever programming language and operating system you prefer.

Several of the examples include a parallel-port interface to a microcontroller circuit. The companion disk has the listings for the microcontroller programs.

**About the Example Circuits**

This book includes schematic diagrams of circuits that you can use or adapt in parallel-port projects. In designing the examples, I looked for circuits that are as easy as possible to put together and program. All use inexpensive, off-the-shelf components that are available from many sources.

The circuit diagrams are complete, with these exceptions:

Power-supply and ground pins are omitted when they are in standard locations on the package (bottom left for ground, top right for power, assuming pin 1 is top left).
Power-supply decoupling capacitors are omitted. (This book explains when and how to add these to your circuits.)
Some chips may have additional, unused gates or other elements that aren't shown.

The manufacturers' data sheets have additional information on the components.

Jameco Part number 138261

Introduction

**Conventions**

These are the typographic conventions used in this book:

| Item | Convention | Example |
|---|---|---|
| Signal name | italics | *Busy, D0* |
| Active-low signal | leading *n* | *nAck, nStrobe* |
| Signal complement | overbar | *C0, S7* (equivalent to *-CO, -S7* or */C0, /S7*) |
| Program code | monospace font | DoEvents, End Sub |
| File name | italics | *win.ini, inpout 16.dll* |
| Hexadecimal number | trailing *h* | 3BCh (same as &h3BC in Visual Basic) |

**Corrections and Updates**

In researching and putting together this book, I've done my best to ensure that the information is complete and correct. I built and tested every circuit and tested all of the program code, most of it multiple times. But I know from experience that on the way from test to publication, errors and omissions do occur.

Any corrections or updates to this book will be available at Lakeview Research's World Wide Web site on the Internet at *http://www.lvr.com*. This is also the place to come for links to other parallel-port information on the Web, including data sheets for parallel-port controllers and software tools for parallel-port programming.

**Thanks!**

Finally, I want to say thanks to everyone who helped make this book possible. I credit the readers of my articles in *The Microcomputer Journal* for first turning me on to this topic with their questions, comments, and article requests. The series I wrote for the magazine in 1994 was the beginning of this book.

Others deserving thanks are product vendors, who answered many questions, and the Usenet participants who asked some thought-provoking questions that often sent me off exploring areas I wouldn't have thought of otherwise.